Category-Driven Association Rule Mining

Zina M. Ibrahim, Honghan Wu, Robbie Mallah and Richard J.B. Dobson

Abstract The quality of rules generated by ontology-driven association rule mining algorithms is constrained by the algorithm's effectiveness in exploiting the usually large ontology in the mining process. We present a framework built around superimposing a hierarchical graph structure on a given ontology to divide the rule mining problem into disjoint subproblems whose solutions can be iteratively joined to find global associations. We present a new metric for evaluating the interestingness of generated rules based on where their constructs fall within the ontology. Our metric is anti-monotonic on subsets, making it usable in an Apriori-like algorithm which we present here. The algorithm categorises the ontology into disjoint subsets utilising the hierarchical graph structure and uses the metric to find associations in each, joining the results using the guidance of anti-monotonicity. The algorithm optionally embeds built-in definitions of user-specified filters to reflect user preferences. We evaluate the resulting model using a large collection of patient health records.

Keywords Association rule mining · Ontologies · Big data

1 Introduction

Ontology-driven association rule mining seeks to enhance the process of searching for association rules with the aid of domain knowledge represented by an ontology. The body of work in this area falls within two themes: (1) using ontologies as models

Z.M. Ibrahim (✉) · H. Wu · R.J.B. Dobson
Department of Biostatistics and Health Informatics, King's College London, London, UK
e-mail: zina.ibrahim@kcl.ac.uk

H. Wu
e-mail: honghan.wu@kcl.ac.uk

R.J.B. Dobson
e-mail: richard.j.dobson@kcl.ac.uk

R. Mallah
The South London and Maudsley NHS Foundation Trust, London, UK
e-mail: robbie.mallah@slam.nhs.uk

M. Bramer and M. Petridis (eds.), *Research and Development in Intelligent Systems XXXIII*, DOI 10.1007/978-3-319-47175-4_2

for evaluating the usefulness of generated rules [3, 8, 9] and (2) using ontologies in a post-mining step to prune the set of generated rules to those that are interesting [10, 12, 13]. In addition to the above, the users in most application domains are usually interested in associations between specific subsets of items in the data. For example, a clinical researcher is almost never interested in associations involving all the articles that appear in her dataset but instead may ask specific queries, such as whether interesting relations exist between medication usage and adverse drug reactions, or the degree of patient conformity to testing procedures and the likelihood of relapse. As a result, many efforts have been directed towards accommodating user preferences given a domain ontology [14–16].

Regardless of the method adopted, the quality of the model is constrained by how well it utilises the (usually) large ontology in the mining process. For example, the Systematized Nomenclature of Medicine Clinical Term Top-level Ontology (SCTTO)[1] has over 100,000 entries. Exploring SCTTO to discover interesting rules from large medical records will yield many possible associations, including irrelevant ones. If SCTTO is used in an association rule mining task, complex queries will be needed to extract the relevant subsets of the ontology. Even then, it is almost inevitable that extensive manual examination is required to maximise relevance.

The above challenges give rise to the need to (1) organise ontology collections to facilitate subsetting and retrieval, (2) build association rule mining algorithms to utilise the organised ontologies to improve the mining process. Our work is motivated by these two needs and revolves around enforcing a meta-structure over an ontology graph. This meta-structure associates a category with a collection of ontology terms and/or relations, creating ontological subcommunities corresponding to categories of interest from a user's perspective. For example, categories may be defined for specific *class of diseases* or *laboratory findings* in SCTTO to investigate novel screening of patients for some disease based on laboratory test results.

This work builds an association rule mining framework which enables the formation of ontology sub-communities defined by categories. The building block of our work is the meta-ontological construct *category*, which we superimpose over domain knowledge and build a representation around. The resulting framework provides (1) translation of user preferences into constraints to be used by the algorithm, to prune domain knowledge and produce more interesting rules, (2) a new scoring metric for rule evaluation given an ontology, and (3) an algorithm that divides rule mining given an ontology into disjoint subproblems whose joint solutions provide the global output, reducing the computational burden and enhancing rule quality. We present a case study of finding associations between the occurrence of drug-related adversities and different patient attributes using hospital records. To our knowledge, meta-ontologies have not been used in conjunction with association rules mining.

[1] https://bioportal.bioontology.org/ontologies/SCTTO

2 Mining Association Rules Revisited

Given a dataset $\mathcal{D} = \{d_1, \ldots, d_N\}$ of N rows with every row containing a subset of items chosen from a set of items $\mathcal{I} = \{i_1, \ldots, i_n\}$, association rule mining finds subsets of $\mathcal{I}$ containing items which show frequent co-occurrence in $\mathcal{D}$.

An association rule is taken to be the following implication: $r : \mathcal{A} \Rightarrow \mathcal{S}$. Where $\mathcal{A}, \mathcal{S} \subseteq \mathcal{I}$ and $\mathcal{A} \cap \mathcal{S} = \emptyset$. $\mathcal{A} = \{i_1, \ldots, i_x\}$ is the set of items of the *antecedent* of an association rule and $\mathcal{S} = \{i_{x+1}, \ldots, i_y\}$ is the set of items of the *consequent* of r. The implication reads that all the rows in $\mathcal{D}$ which contain the set of items making up $\mathcal{A}$ will contain the items in $\mathcal{S}$ with some probability Pr.

Two measures establish the strength of an association rule: *support* and *confidence*. *Support* Determines how often a rule is applicable to a given data set and is measured as the probability of finding rows containing all the items in the antecedent and consequent, $Pr(\mathcal{A} \cup \mathcal{S})$, or $\frac{|\mathcal{A} \cup \mathcal{S}|}{N}$, where N is the total number of rows. *Confidence* determines how frequently items in $\mathcal{S}$ appear in rows containing $\mathcal{A}$ and is interpreted as the conditional probability $Pr(\mathcal{S}|\mathcal{A})$. Therefore, *support* is a measure of statistical significance while *confidence* is a measure of the strength of the rule. The goal of association rule mining is to find all the rules whose *support* and *confidence* exceed predetermined thresholds [17].

Support retains a useful property which states that the support of a set of items never exceeds the support of its subsets. In other words, *support* is anti-monotonic on rule subsets. More specifically, let r_1 and r_2 be two association rules where $r_1 : \mathcal{A}_1 \Rightarrow \mathcal{S}_1$ and $r_2 : \mathcal{A}_2 \Rightarrow \mathcal{S}_2$, then the following holds [1]:

$$\mathcal{A}_1 \cup \mathcal{S}_1 \subseteq \mathcal{A}_2 \cup \mathcal{S}_2 \rightarrow support(\mathcal{A}_1 \cup \mathcal{S}_1) \geq support(\mathcal{A}_2 \cup \mathcal{S}_2)$$

Althrough *confidence* does not adhere to general anti-monotonicity, the confidence of rules generated using the same itemset is anti-monotonic with respect to the size of the consequent, e.g. if $\mathcal{I}_s \subset I$ is an itemset such that $\mathcal{I}_s = \{A, B, C, D\}$, all rules generated using all elements of $\mathcal{I}_s$ will be anti-monotonic with respect to the consequents of the possible rules. e.g. $confidence(ABC \rightarrow D) \geq confidence(AB \rightarrow CD) \geq confidence(A \rightarrow BCD)$.

Unlike in *confidence*, the anti-monotonicity of *support* is agnostic to the position of the items in the rule (i.e. whether they fall within the antecedent or the consequent). Therefore, when evaluating *support*, a rule is collapsed to the unordered set of items $\mathcal{A} \cup \mathcal{S}$. This difference has been exploited by the Apriori algorithm [1] to divide the association mining process into two stages: (1) a candidate itemset generation stage aiming to reduce the search space by using *support* to extract unordered candidate itemsets that pass a predetermined frequency threshold in the dataset, and (2) a rule-generation stage which discovers rules from the frequent itemsets and uses *confidence* to return rules which pass a predetermined threshold. In both stages, anti-monotonicity prevents generating redundant itemsets and rules by iteratively generating constructs of increasing lengths and avoiding the generation of supersets

that do not pass the *support* and *confidence* thresholds [1]. Our work uses these principles to build a category-aware ontology-based Apriori-like framework.

3 Category-Augmented Knowledge

The building block of this work is the meta-ontological construct *category* we use to augment an ontology. Let $\mathcal{K} = (\mathcal{O}, \mathcal{R})$ be our knowledge about some domain defined by a set of terms (classes and instances) $\mathcal{O} = \{o_1, \ldots, o_n\}$, also called the universe, and a set of relations $\mathcal{R} = \{r_1, \ldots, r_k\}$ connecting the elements of $\mathcal{O}$. Moreover, let $\mathcal{C} = c_1, \ldots, c_m$ be a non-empty set of categories describing different groups to which the elements of $\mathcal{O}$ belong, such that $m << n$. The basic idea is to superimpose $\mathcal{C}$ over $\mathcal{O}$, creating subcommunities in the ontology graph which can be processed individually. To achieve this, we first define a mapping from $\mathcal{O}$ to $\mathcal{C}$ which organises the elements of $\mathcal{O}$ into subcommunities. The intuition is that every category in $\mathcal{C}$ represents a group of interest which can be mined for associations individually or in conjunction with other groups.

We can therefore define a mapping $\mathcal{F} : \mathcal{O} \times \mathcal{C} \rightarrow \{0, 1\}$ to yield a value of 1 whenever a concept $o \in \mathcal{O}$ is associated with the category $c \in \mathcal{C}$, and 0 otherwise. $\mathcal{F}$ is exhaustive over $\mathcal{O}$, i.e. every element in the universe must belong to a category. Formally: $\forall o \in \mathcal{O}, \exists c \in \mathcal{C}$ such that $\mathcal{F}(o, c) = 1$.

A function $\sigma : \mathcal{C} \rightarrow \mathcal{O}$ can then be defined to extract the set of elements in the universe associated with a category $c \in \mathcal{C}$:

$$\sigma(c) = \mathcal{O}_c \subset \mathcal{O} : \forall o_c \in \mathcal{O}_c, \mathcal{F}(o_c, c) = 1 \tag{1}$$

Because $\mathcal{F}$ is exhaustive, the inverse of σ is also a function. $\sigma^{-1} : \mathcal{O} \rightarrow \mathcal{C}$ yields the set of categories to which an element o belongs (an element may belong to multiple categories):

$$\sigma^{-1}(o) = \mathcal{C}_o \subset \mathcal{C} \iff \forall c_o \in \mathcal{C}_o : \sigma(c_o) = o \tag{2}$$

3.1 Graphical Representation

To represent category-augmented background knowledge (ontology) graphically, we borrow the concept of a *hierarchical graph* [5], which is one whose nodes may contain other graphs and arcs can contain other arcs. The graph contained in a node is called a *subgraph* of that *parent node*. The arcs that connect two nodes belonging to the same subgraph are called *internal arcs*, while arcs connecting nodes in different subgraphs of the same hierarchical level are called *external arcs*, and the nodes that are connected in that way are called *border nodes* of their respective subgraphs. No arc is allowed to connect two nodes of different hierarchical levels.

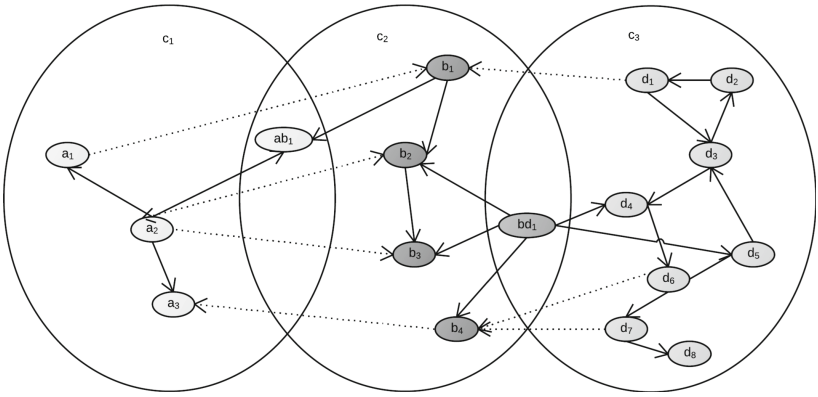

Fig. 1 A two-tier hierarchical graph

To capture the properties of a category-augmented ontology as described earlier, we define a two-tier hierarchical graph structure such as the one shown in Fig. 1. In the figure, the three subgraphs correspond to three categories c_1, c_2 and c_3. The solid arcs are internal to each subgraph while the dotted arcs are the external arcs of the graph. A formal definition of a two-tier hierarchical graph follows.

Definition 1 Let $\mathcal{K} = (\mathcal{O}, \mathcal{R})$ be some domain knowledge, and let $\mathcal{C} = \{c_1, \ldots, c_n\}$ be a set of categories such that $\mathcal{F} : \mathcal{O} \times \mathcal{C} \rightarrow \{0, 1\}$ is defined. A two-tier hierarchical graph $\mathcal{G} = (\mathcal{V}(\mathcal{G}), \mathcal{E}(\mathcal{G}))$ represents $\mathcal{K}$ with $\mathcal{C}$ superimposed such that:

1. Nodes in $\mathcal{V}(\mathcal{G})$ are subgraphs connecting subsets of the universe belonging to a single category $c \in \mathcal{C}$. We denote the elements of $\mathcal{V}(\mathcal{G})$ by **tier-one** nodes characterise them as follows:
   a. The number of tier-one nodes corresponds to the number of categories in $\mathcal{C}$, i.e. $\mathcal{V}(\mathcal{G})| = |\mathcal{C}|$
   b. The subgraphs corresponding to the nodes in $\mathcal{V}(\mathcal{G})$ comprise internal nodes which are a subset of the universe associated with the category, and arcs that are a subset of $\mathcal{E}(\mathcal{G})$ connecting the internal nodes, i.e. $\forall c \in \mathcal{C}, \exists \mathcal{G}_c \in \mathcal{V}(\mathcal{G})$ such that $\mathcal{G}_c = (\mathcal{V}(\mathcal{G}_c), \mathcal{E}(\mathcal{G}_c))$ corresponds to a subgraph of $\mathcal{G}$ given category c with $\mathcal{V}(\mathcal{G}_c)$ as nodes and $\mathcal{E}(\mathcal{G}_c)$ as its set of arcs further defined as follows:
   - The nodes in each subgraph $\mathcal{V}(\mathcal{G}_c)$ is the subset of $\mathcal{O}$ associated with c:

$$\mathcal{V}(\mathcal{G}_c) = \sigma(c)$$

   These nodes are termed the **tier-two** nodes of the graph.
   - The set of arcs $\mathcal{E}(\mathcal{G}_c)$ is mapped from a subset of the set of relations $\mathcal{R}$ which only contains the relations connecting universe elements exclusively associated with c. For any two nodes $o_1, o_2 \in \mathcal{O}$:

$$\forall e_c = (o_1, o_2) \in \mathcal{E}(\mathcal{G}_c), \;\; c \in \sigma^{-1}(o_1) \wedge c \in \sigma^{-1}(o_2)$$

2. $\mathcal{E}(\mathcal{G})$ is the set of external arcs and connects the different subgraphs by connecting their corresponding border nodes as dictated by $\mathcal{K}$ and $\mathcal{C}$:

   - $\forall e \in \mathcal{E}(\mathcal{G})$, e connects two subgraphs associated with categories c_1 and c_2 if $\exists r \in \mathcal{R}$ such that r connects two nodes o_1 and $o_2 \in \mathcal{O}$ which exclusively belong to the respective categories. In other words o_1 and o_2 satisfy:

$$c_1 \in \sigma^{-1}(o_1) \wedge c_1 \notin \sigma^{-1}(o_2) \wedge c_2 \in \sigma^{-1}(o_2) \wedge c_2 \notin \sigma^{-1}(o_1)$$

In Fig. 1, $\mathcal{G}$ is defined by the tier-one nodes $\mathcal{V}(\mathcal{G}) = \{c_1, c_2, c_3\}$ and external arcs $\mathcal{E}(\mathcal{G})$ shown as dotted lines. Each element of $\mathcal{V}(\mathcal{G})$ is in turn a subgraph $\mathcal{G}_c$ containing the nodes within the subgraph (tier-two nodes) and the solid arcs in each subgraph, i.e. $\mathcal{V}(\mathcal{G}) = \{\mathcal{G}_{c_1}, \mathcal{G}_{c_2}, \mathcal{G}_{c_3}\}$. Moreover, $\mathcal{V}(\mathcal{G})_{c_1} = \{a_1, a_2, a_3, ab_1\}$, $\mathcal{V}(\mathcal{G})_{c_2} = \{b_1, b_2, b_3, b_4, ab_1, bd_1\}$ and $\mathcal{V}(\mathcal{G})_{c3} = \{d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8, bd_1\}$. Note that ab_1 is shared between subgraphs c_1 and c_2 and bd_1 is shared between subgraphs c_2 and c_3, reflecting that an element in the universe may belong to more than one category. A similar observation can be made for the other two subgraphs.

4 Category-Augmented Rule-Association Mining Using Background Knowledge

Let $\mathcal{K} = (\mathcal{O}, \mathcal{R})$ be our ontology with $\mathcal{C}$ categories as before. Let $\mathcal{G} = (\mathcal{V}(\mathcal{G}), \mathcal{E}(\mathcal{G}))$ be a two-tier hierarchical graph representation of $\mathcal{C}$ superimposed on $\mathcal{O}$. Let $\mathcal{D} = \{d_1, \ldots, d_N\}$ be a data set of N records, where each row $d_i \in \mathcal{D}$ contains a subset of items chosen from a predefined set of items $\mathcal{I} = \{i_1, \ldots, i_n\}$. Every element of $\mathcal{I}$ corresponds to a node in $\mathcal{O}$. To represent this, we define a one-to-one and onto mapping $\mathcal{M} : \rightarrow \mathcal{O}$ which maps each item in $\mathcal{I}$ to a node in $\mathcal{O}$.

4.1 Category-Derived Constraints

The category-augmented knowledge framework introduced so far can be used to define constraints on the association rules to be discovered. We can use the constraints to determine user preferences to guide the algorithm to avoid performing an unnecessary search. Given a dataset $\mathcal{D}$, we define four types of rule constraints:

Definition 2 Let $\mathcal{K} = (\mathcal{O}, \mathcal{R})$ our domain knowledge with $\mathcal{C} = \{c_1, \ldots, c_m\}$ being the set of categories superimposed over $\mathcal{O}$ as before ($m << n$). Let $r : \mathcal{A} \Rightarrow \mathcal{S}$ be an association rule with antecedent $\mathcal{A}$ and consequent $\mathcal{S}$ where $\mathcal{A} = \{i_1, \ldots, i_x\} \subseteq \mathcal{I}$ and $\mathcal{S} = \{i_{x+1}, \ldots, i_y\} \subseteq \mathcal{I}$. Moreover, let the mapping $\mathcal{M} : \mathcal{I} \rightarrow \mathcal{O}$ hold and let $\mathcal{C}_p \subseteq \mathcal{C}$ be a subset of the categories imposed on $\mathcal{O}$.

1. r is said to adhere to a head-inclusion constraint on $\mathcal{C}_p$ if all the items in its antecedent map to concepts in $\mathcal{O}$ which are associated with a category which falls within $\mathcal{C}_p$.
$$\forall i \in \mathcal{A}(r) : \mathcal{M}(i) = o \wedge \sigma^{-1}(o) \subset \mathcal{C}_p$$

2. r is said to adhere to a head-exclusion constraint on $\mathcal{C}_p$ if none of the items in its antecedent map to concepts in $\mathcal{O}$ which are associated with a category which falls within $\mathcal{C}_p$.
$$\neg\exists i \in \mathcal{A}(r) : \mathcal{M}(i) = o \wedge \sigma^{-1}(o) \subset \mathcal{C}_p$$

Tail-inclusion and tail-exclusion constraints are similarly defined by replacing $\mathcal{A}$ with $\mathcal{S}$ in points 1 and 2 respectively.

4.2 Score Evaluation

We would like to use a scoring function that can (1) accommodate both $\mathcal{D}$ and the ontology represented by $\mathcal{G}$, and (2) retain monotonicity on the model so that the Apriori principle [1] can be used. Therefore, we formulate *interest*, a scoring metric that measures how interesting a rule is given an ontology by quantifying the goodness of the fit between the two. *interest* is based on the following two components:

1. The lengths of the paths connecting two-tier nodes that correspond to items in $\mathcal{A} \cup \mathcal{S}$. Shorter paths reflect more direct relationships and are more likely to form interesting associations. We define the distance between two tier-two nodes $d(o_i, o_j)$ as the length of the shortest undirected path connecting them. To express our preference to shorter paths, we use the ratio of between the minimum distance connecting any two tier-two nodes in the graph to $d(o_i, o_j)$. The resulting measure ζ quantifies the interestingness of the relations among $\mathcal{A} \cup \mathcal{S}$ items by the sum of their pairwise distance ratios. $\zeta : \mathcal{O} \rightarrow [0 - 1]$ is defined below.
$$\zeta(\mathcal{O}_k|\mathcal{G}) = \min_{o_i, o_j \in \mathcal{O}_k} d(o_i, o_j) \times \sum_{o_i, o_j \in \mathcal{O}_k} \frac{1}{d(o_i, o_j)}, \mathcal{O}_k \subseteq \mathcal{O}$$

2. The degrees of the nodes reflect their centrality within the graph, which we use as a reciprocal of interestingness. The hypothesis is that more significant relations exist among nodes which connect to fewer other nodes. For instance, in the worst case scenario where a tier-two node o connects to every other node in the graph, no information is gained from finding an association translating to (o, o_i) in the data (with o_i being any other tier-two node).

   We define the degree as the number of undirected relations the node forms within the graph in question. The definition of the degree is context-specific, i.e. the degree of a node can be different depending on whether it is computed relative to

the entire graph or the one induced by a given category:

$$deg(o|\mathcal{G}) = |\mathcal{E}_o|, \mathcal{E}_o \subset \mathcal{E}(\mathcal{G}), \forall e \in \mathcal{E}_o : e = (o_i, o) \vee e = (o, o_i)$$

where o_i is any other tier-two node in the graph. The reader should note that when $\mathcal{G}$ is taken as the subgraph induced by a category, then $\mathcal{E}(\mathcal{G})$ will correspond to the arcs internal to the graph, according to the definition of tier-one nodes being subgraphs of specific categories (Definition 1). Therefore the external arcs will not count towards the degree. This results in a value corresponding to the degree of the node relative to the internal structure of the graph induced by the category. Having defined the degree, we can now determine degree-based interestingness of a set of tier-two nodes given a graph as the sum of the reciprocals of their respective degrees within the graph. $\psi : \mathcal{O} \rightarrow \mathbb{R}_+$ is defined as:

$$\psi(\mathcal{O}_k|\mathcal{G}) = \sum_{o_i \in \mathcal{O}_k} \frac{1}{deg(o_i|\mathcal{G})}, \mathcal{O}_k \subseteq \mathcal{O}$$

We can now define the *interest* of a set of items given an ontology graph $\mathcal{G}$ as:

Definition 3 Let $\mathcal{G} = (\mathcal{V}(\mathcal{G}), \mathcal{E}(\mathcal{G}))$ be a two-tier hierarchical graph representation of domain knowledge $\mathcal{K} = (\mathcal{O}, \mathcal{R})$. Let $r : \mathcal{A} \Rightarrow \mathcal{S}$ be a rule with $\mathcal{A} \cup \mathcal{S}$ map to a collection of items from the universe and are represented by tier-two nodes in $\mathcal{V}(\mathcal{G})$. The *interest* of r given $\mathcal{G}$ is:

$$interest(r|\mathcal{G}) = \sqrt{\zeta(\mathcal{A} \cup \mathcal{S}) \times \psi(\mathcal{A} \cup \mathcal{S})}$$

Proposition 1 *interest(r|$\mathcal{G}$) is anti-monotonic with respect to subsets. Formally, let* $r_1 : \mathcal{A}_1 \Rightarrow S_1$ *and* $r_2 : \mathcal{A}_2 \Rightarrow S_2$ *be two rules, then:* $\mathcal{A}_1 \cup \mathcal{S}_1 \subseteq \mathcal{A}_2 \cup \mathcal{S}_2 \rightarrow interest(\mathcal{A}_1 \cup \mathcal{S}1_) \geq interest(\mathcal{A}_2 \cup \mathcal{S}_2)$

Justification 1 *Given that* $\mathcal{A}_1 \cup \mathcal{S}_1 \subseteq \mathcal{A}_2 \cup \mathcal{S}_2$ *and* $\zeta : \mathcal{V}(\mathcal{G}) \rightarrow [0 - 1]$, *this implies that* $\zeta(\mathcal{A}1 \cup \mathcal{S}_1) \geq \zeta(\mathcal{A}_2 \cup \mathcal{S}_2)$ *because* $|\mathcal{A}_1 \cup \mathcal{S}_1| \leq |\mathcal{A}_2 \cup \mathcal{S}_2|$.

Similarly, because $|\mathcal{A}_1 \cup \mathcal{S}_1| \leq |\mathcal{A}_2 \cup \mathcal{S}_2|$, *then* $\psi(\mathcal{A}_1 \cup \mathcal{S}_1) \geq \psi(\mathcal{A}_2 \cup \mathcal{S}_2)$. *It follows that:* $interest(\mathcal{A}_1 \cup \mathcal{S}_1) \geq interest(\mathcal{A}_2 \cup \mathcal{S}_2)$.

4.3 The Algorithm

The algorithm presented here relies on a two-tier hierarchical graph and the *interest* scoring metric to mine rules from a given dataset. Category-miner is a modified Apriori algorithm and consists of the two Apriori stages: (1) a candidate itemsets generation stage, which uses our *interest* metric in addition to *support* to generate itemsets that pass the frequency test and reflect a good fit with the ontology, (2) a rule generation step, using the candidate itemsets (stage 1) to generate rules that pass the

confidence threshold. The algorithm also considers user preferences by incorporating the category-derived constraints we defined in Sect. 4.1.

Algorithm 1 is a wrapper algorithm. It receives as input a two-tier hierarchical graph $\mathcal{G}$ representing an ontology augmented with categories $\mathcal{C}$ and a dataset $\mathcal{D}$, where $\mathcal{D}$ maps to nodes in the ontology and will be used to generate candidate itemsets. The four optional parameters hi, he, ti and te correspond to the four category-derived constraints (Sect. 4.1) and are used to specify user preferences.

Algorithm 1 Association Rule Mining Wrapper Algorithm

Input: $\mathcal{G}, \mathcal{D}, hi, he, ti, te$
Output: Set of association rules $\mathcal{R}$
Procedure:
1: $\mathcal{S} \leftarrow \emptyset$ // set of candidate itemsets, initially empty
2: $\mathcal{HS} \leftarrow (\mathcal{C} \setminus he)$; $\mathcal{TS} \leftarrow (\mathcal{C} \setminus te)$
3: $\mathcal{HS} \leftarrow \mathcal{HS} \cap hi$ **if** $hi \neq \emptyset$
4: $\mathcal{TS} \leftarrow \mathcal{TS} \cap ti$ **if** $ti \neq \emptyset$
5: **for** $c_i \in \mathcal{HS} \cup \mathcal{TS}$ **do**
6: $\mathcal{S} \leftarrow \mathcal{S} \cup$ category-miner($\mathcal{G}$,$\mathcal{D}$,c_i)
7: $\mathcal{S} \leftarrow$ expand($\mathcal{S}, \mathcal{G}$)
8: $\mathcal{R} \leftarrow$generate-rules($\mathcal{S}, \mathcal{HS}, \mathcal{TS}$)
9: **Return:** $\mathcal{R}$

The wrapper algorithm constrains the antecedent and consequent by user preferences (lines 2–4). The variables head set ($\mathcal{HS}$) and tail set ($\mathcal{TS}$) contain all the categories permissible in the antecedent and consequent of any generated rule respectively. $\mathcal{HS}$ and $\mathcal{TS}$ are generated by excluding the categories specified in the exclusion constraints (*he* and *te* respectively) from $\mathcal{C}$ and if inclusion sets are provided (*hi* or *ti*), they will be the only sets used in the. Providing empty hi, he, ti and te sets makes our procedure equivalent to the general (non-constrained) algorithm.

The algorithm iteratively calls category-miner (lines 5–6), which generates candidate itemsets whose constructs fall strictly within the same category, for every category c_i in $\mathcal{HS}$ and $\mathcal{TS}$. This stage is agnostic to the position of the items in the rule. Hence category-miner is called for all categories in $\mathcal{HS} \cup \mathcal{TS}$ (line 6).

In category-miner (Algorithm 2), the initial 1-item itemsets are only pruned using *support* (line 2) because *interest* evaluates relationships rather than objects (requiring at least two-item itemsets). candidate-gen (line 4) generates all k-itemset supersets of a k -1-itemsets. The results are pruned at every iteration on the data using *support* and on the ontology using *interest*. Anti-monotonicity of the two metrics guarantees correctness, ensuring that any interesting and supported k-itemsets are composed using interesting and supported $k-1$-itemsets.

Once candidate itemsets associated with all categories are generated, an informed search is performed for supersets which transcend the category boundaries using the expand procedure (Algorithm 3). The algorithm uses anti-monotonicity once again to formulate the hypothesis: since all within-category associations have been found, the rules spanning the categories can be identified by evaluating the scores of

their supersets. These supersets are found by examining the external arcs $\mathcal{E}(\mathcal{G})$ and adding their connecting nodes to the existing sets if they result in associations which pass our *support* and *interest* tests. For each external arc (o_i, o_j) used for expansion search, we obtain the set of associations previously found which strictly contain node o_i (line 2) and the set of associations which were previously found to strictly contain node o_j (line 3). Supersets are found by examining pair-wise unions of the generated sets which pass the goodness test (lines 5–6).

Algorithm 2 Category-specific Mining (category-miner)

Input: $\mathcal{G}, \mathcal{D}, \mathcal{S}$
Output: Subset of associations for the category $\mathcal{S}$
Procedure:
1: $//\mathcal{S}_1$ contains interesting 1-itemset from the category subgraph
2: $\mathcal{S}_1 \leftarrow o \in \mathcal{S},\ \ support(o|\mathcal{D}) \geq t_1$
3: **for** $(k = 2; \mathcal{S}_{k-1} \neq \emptyset; k++)$ **do**
4: $\quad \lambda_k = candidate - gen(\mathcal{S}_{k-1}, \mathcal{D})$ // New Candidate set
5: $\quad \mathcal{S}_k = \{s \in \lambda_k | support(s|\mathcal{D}) > t_1; interest(s|\mathcal{G}) > t_2\}$
6: $\quad$ *remove-proper-subsets*$(\mathcal{S}_k)$
7: **Return:** $\bigcup_k \mathcal{S}_k$

Algorithm 3 Itemset Expansion Algorithm (expand)

Input: $\mathcal{S}, \mathcal{G}$
Output: Inter-category rule set
Procedure:
1: **for** $e = (o_i, o_j) \in \mathcal{E}_\mathcal{G}$ **do**
2: $\quad \mathcal{S}_{o_i} \subset \mathcal{S} : \forall s_{o_i} \in \mathcal{S}_{o_i}, o_i \in s_{o_i} \wedge o_j \notin s_{o_i}$
3: $\quad \mathcal{S}_{o_j} \subset \mathcal{S} : \forall s_{o_j} \in \mathcal{S}_{o_j}, o_j \in s_{o_j} \wedge o_i \notin s_{o_j}$
4: $\quad \mathcal{S}_{ij} = \{s_{o_i} \cup s_{o_j} | s_{o_i} \in \mathcal{S}_{o_i}, s_{o_j} \in \mathcal{S}_{o_j}\}$
5: $\quad \mathcal{S} \leftarrow \{\mathcal{S} \cup \mathcal{S}_{ij} | support(\mathcal{S}_{ij}|\mathcal{D}) > t_1; interest(\mathcal{S}_{ij}|\mathcal{G}) > t_2)\}$
6: **Return:** $\mathcal{S}$

The expand procedure also uses the anti-monotonicity to guarantee correctness. The unions are performed pairwise for every pair of itemsets connected with an external arc in the ontology, which (1) makes the procedure complete by not losing interesting associations as all pairs are considered, and (2) reduces the computational burden by only considering itemset pairs with nodes sharing an external arc (lines 2–3). The resulting itemset $\mathcal{S}$ is returned to the wrapper algorithm (line 10), which marks the end of the itemset generation step. The rule generation step (line 11 of wrapper) is not shown as it is similar to Apriori's rule generation, with additional pruning according to user preferences. *confidence* is used iteratively to generate rules from $\mathcal{S}$ and the final set of rules $\mathcal{R}$ is returned by the algorithm.

4.4 Time Complexity

Given our database $\mathcal{D}$ with N rows and n items in each row, let m be the number of categories such that $n >> m$. cateogry-miner receives as input $M = N \times \frac{n}{m}$ items on average (the number of elements in each row in the category is $\frac{n}{m}$.

In category-miner, the first iteration evaluating 1-item sets requires M comparisons. Subsequently, for each iteration k generating $|\mathcal{R}_k|$ candidate itemsets, the complexity will be $\mathcal{O}(|\mathcal{R}_{k-1}|.|\mathcal{R}_k|)$ because each iteration will compare to the itemsets generated in the previous step. The resulting complexity will be $k \times \sum_k \mathcal{O}(|\mathcal{R}_{k-1}|.|\mathcal{R}_1|)$. The worst time complexity occurs when only one category is imposed on the ontology, resulting a running time equivalent to that of the Apriori algorithm with the added computations induced by calculating *interest*. Moreover, the worst-case time complexity of the *expand* procedure is $|\mathcal{E}_\mathcal{G}|.|\mathcal{S}|^2$, which is only reached in the unlikely situation where every node in every subgraph shares an external arc with another node in another subgraph.

The performance gain in our algorithm is due to the use of multiple categories, which prevents the worst case of category-miner by ensuring that M is small compared to the total number of items that would have been supplied to the algorithm if no categories were enforced (which is $n \times N$), and delegating more to the expand procedure. Figure 2 shows how the number of comparisons is reduced by increasing the number of categories supplied to the algorithm using 2–7 itemsets.

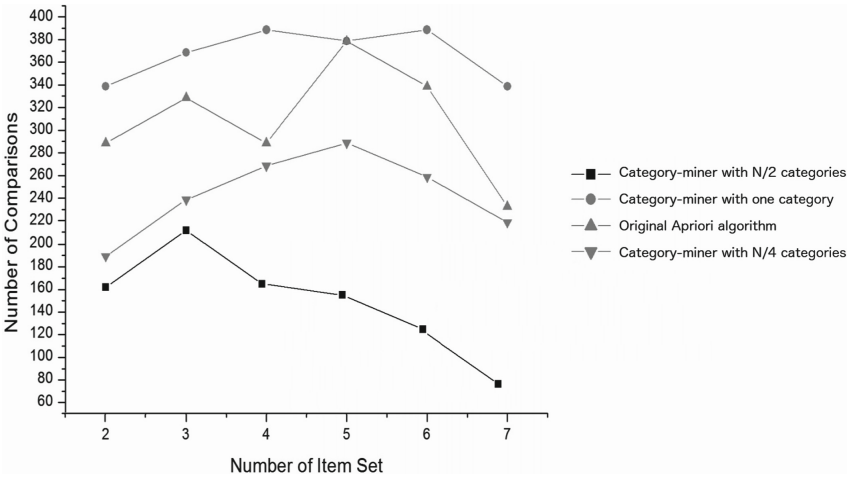

Fig. 2 Number of comparisons under different settings

5 Experimental Evaluation Using Medical Records

We now describe the process of selecting association rules from a large database of anonymised hospital medical records for 253 patients with bipolar disorder in the South London and Maudsley NHS Foundation Trust. We extracted a dataset of patient records containing $|\mathcal{I}| = 675$ items corresponding to six categories which we selected with the help of pharmacists: symptoms, medications, co-morbidities (diseases in addition to bipolar), compliance indicators, patient characteristics such as height and weight and finally occurrences of adverse drug reactions. We used UMLS (Unified Medical Language System) as a reference ontology to extract a subset containing our 675 items and superimposed a hierarchical graph on it using the six categories. At the time of writing, the task of constructing the two-tier graph is done in a semi-automated way where experts take the task of assigning class nodes to a set of predefined categories used to construct the graph. The procedure assigns instance nodes the categories of their respective class nodes. The relations are mapped into arcs with the semantics retained. We ran this procedure on the subset of UMLS we extracted for the six categories. We designed three experiments to test our framework and selected the top 30 % rules/items ranked by the scores.

1. **Evaluating the scoring metric**: To evaluate *interest*, we compared category-miner to the standard Apriori algorithm for each of the six categories individually. Since the difference between category-miner and the Apriori algorithm is the addition of the *Interest* metric, the quality of the rules accepted by the Apriori algorithm and rejected by category-miner reflects the goodness of our metric.
   We first compute the rate of rejecting rules generated by Apriori using *interest*. This is necessary because only high rates of rejection justify evaluating rule quality (low rates imply comparable performance of the two algorithms). The overall rejection rate was 42 %, distributed among the categories (Table 1).

   Second, we evaluated the quality of the rejected rules using the help of pharmacists. In total, 236 of the 257 rejected rules were found uninteresting by the experts (true negatives, 92 %) and only 21 were found to be interesting (false negatives, 8 %). Among 354 rules which were accepted by both algorithms, 203 were found interesting (true positives, 57 %) and 151 uninteresting (false positives, 43 %). Therefore, we can conclude that the strength of our approach lies in filtering the generated rules by rejecting ones with accepted *support* which do not agree with

Table 1 Rules generated by the Apriori versus Category-miner. C: patient characteristics, H: patient habits, M: medications, B: comorbidities, S: symptoms and A: adverse drug reactions.

Category	C	H	M	B	S	A
N. Rule (Apriori)	59	46	61	53	70	65
N. Rule (category-miner)	80	88	99	112	105	127
Rejection rate %	26.3	47.7	38.4	52.7	33.3	48.8

the ontology. It is, however, worth noting that upon manual examination, many of the rules accepted by our algorithm correspond to hierarchical and instance-of relations (e.g. drug - clozapine) which can be filtered by discarding them from the algorithm's search space. These filters are part of our ongoing work.

2. **Comparison with Other Ontology-driven Approaches** We also compare our results to knowledge-based approaches. As a model, we chose [8] as it is conceptually similar to our approach and formulates an ontology-based probabilistic measure to evaluate the generated rules. We ran both algorithms using the full set of categories (using as an ontology the full subset of UMLS extracted). The resulting set of rules includes 588 rules accepted by both approaches and 287 rules rejected by both approaches. 124 rules were rejected by our approach but not that of [8] and 111 rules were rejected by [8] and accepted by our approach. We first asked pharmacists to assess the quality of the rejected rules. The results were 85 % true negatives (and 15 % false negatives) using our algorithm versus 76 % true negatives (and 24 % false negatives using [8]). The rules jointly rejected by both approaches had 93 % true negatives (and 7 % false negatives).

   With respect to the accepted rules, those accepted by both algorithms reported 63 % true positives. The rules accepted by our approach reported 68 % true positives while the rules accepted by [8] reported 52 % true positives. Despite being significantly higher than the rate reported by [8], these results do not show conclusive improvement by our algorithm. However, once again, manual examination shows that 37 % of the false positives reported by our algorithm correspond to class hierarchy and instance-of relations which we have not accommodated in the filtration step. Therefore, we anticipate higher power than what is being reported here. The reader should note that the work in [8] accommodates the existence of is-a relations and considers them uninteresting. As a result, they do not appear in their results. In addition to these findings, the results of this experiment show a higher true positive rate than the first experiment. This shows that many of the rules found interesting by experts are in fact the inter-category rules.
3. **Evaluating the Constrained Algorithm** The last experiment evaluates the effectiveness of the constraints in producing rules which are interesting to the user. The experiment was conducted by restricting the categories permissible in the antecedents to comorbidities and medications and those permissible in the consequents to adverse drug reactions. Clinicians were asked whether the rules generated reflect known joint associations between co-morbidities, medications and the possibility of developing adverse drug reactions. 154 rules were generated by the constrained algorithm, among which 53 were true positive, and 12 were false positive. The constrained algorithm shows better rates than the general one, which can be attributed to the reduced search space but also shows the appropriateness of the definitions of constraints we presented here.

6 Summary and Conclusions

We used *categories* to superimpose a structure on ontologies and define a framework to decompose the set of rules to be generated into disjoint subsets and joined iteratively at a later step. We evaluated the framework for performance and quality of output using an extensive database of patient records.

We are currently developing methods for evaluating the relations between items to discard trivial hierarchical, instance and property relations from being used to generate itemsets and rules. The idea of decomposing the mining problem into disjoint subsets is part of a large effort to create parallel algorithms which can operate on very large and possibly distributed data sources.

Acknowledgments The authors would like to acknowledge the National Institute for Health Research (NIHR) Biomedical Research Centre and Dementia Unit at South London and Maudsley NHS Foundation Trust and Kings College London.

References

1. Agrawal, R., Imielinski, T., Swami, A.: *Mining association rules between sets of items in large databases*. ACM SIGMOID. 207-216. ACM Press (1993)
2. Baclawski, K., Schneider, T.: The open ontology repository initiative: requirements and research challenges. In: Workshop on Collaborative Management of Structured Knowledge at the ISWC (2009)
3. Cherfi, H., Napoli, A., Toussaint, Y.: Towards a text mining methodology using association rule extraction. Int. J. Soft Comput. **10**, 431–441 (2006)
4. Cornet, R., de Keizer, N.: Forty years of SNOMED: a literature review. BMC Med. Inf. Decis. Making **8**(1), S2 (2008)
5. Fernandez, J., Gonzalez, J.: Hierarchical graph search for mobile robot path planning. In: IEEE ICRA, pp. 656–661 (1998)
6. Garbacz, P., Trypuz, R.: A metaontology for applied ontology. Appl. Ontol. **8**, 1–30 (2013)
7. Herre, H., Loebe, F.: A meta-ontological architecture for foundational ontologies. On the Move to Meaningful Internet Systems, pp. 1398–1415 (2005)
8. Janetzko, D., Cherfi, H., Kennke, R., Napoli, A., Toussaint, Y.: Knowledge-based selection of association rules for text mining. In: ECAI, pp. 485–489 (2004)
9. Lieber, J., Napoli, A., Szathmary, L.: First elements on knowledge discovery guided by domain knowledge. Concept Lattices Appl. **4923**, 22–41 (2008)
10. Marinica, C., Guillet, F.: Knowledge based interactive postmining of association rules using ontologies. IEEE Trans. Knowl. Data Eng. **22**(6), 784–797 (2010)
11. Palma, P., Hartmann, J., Haase, P.: Ontology metadata vocabulary for the semantic web. Technical Report, Universidad Politcnica de Madrid, University of Karlsruhe (2008)
12. Ramesh, C., Ramana, K., Rao, K., Sastry, C.: Interactive post-mining association rules using cost complexity pruning and ontologies KDD. Int. J. Comput. Appl. **68**(20), 16–21 (2013)
13. Savasere, A., Omiecinski, E., Navathe, S.: An efficient algorithm for mining association rules in large databases. In: VLDB, pp. 432–444 (1995)
14. Singh, L., Chen, B., Haight, R., Scheuermann, P.: An algorithm for constrained association rule mining in semi-structured data. In: PAKDD, pp. 148–158 (1999)
15. Song, S., Kim, E., Kim, H., Kumar, H.: Query-based association rule mining supporting user perspective. Computing **93**, 1–25 (2011)

16. Srikant, R., Vu, Q., Agrawal, R.: Mining association rules with item constraints. In: KDD, pp. 67–73 (1997)
17. Tan, P., Steinbach, M.: *Introduction to Data Mining*. Addison-Wesley (2006)